

Modulname	Python Programmierung 1		
Lead	<p>Python entwickelt sich zu einer End-User Programmiersprache auch für Nicht-Informatiker. So kann aus Python z.B. direkt auf Excel-Daten (nicht nur csv-Daten) zugegriffen werden, Objekte mittels KI aus einem Videostream erkannt oder mit dem RaspberryPi (RPI) (Linux Microcontroller System) Maschinen-Steuerungen programmiert werden. In Python können sie in wenigen Zeilen Code einen Webservice (SOAP oder REST) aufrufen, die Response verarbeiten oder einen eigenen Service mit verschiedenen End-Points implementieren. In diesem Kurs setzen Sie ihren eigenen RaspberryPi auf und lernen die Basics von Python kennen. Am Schluss sind Sie in der Lage, Command-Line (CLI) Applikationen in Python auf dem RPi zu implementieren, welche auch Sensordaten verarbeitet, Aktoren steuern und REST Services aufrufen.</p>		
Zielgruppe	Studierende der Höheren Fachschule in den Fachrichtungen Automation, Elektrotechnik, Informatik und Erneuerbare Energie		
Inhalt	<p>Eigener RaspberryPi in Betrieb nehmen- Hardware und GPIO</p> <ul style="list-style-type: none"> - WiFi Setup - Sense-Hat - Linux image erstellen - Entwicklungs-Umgebung auf BYOD einrichten (PyCharm oder VSC) <p>Grundlagen von Python- Linux: Filesystem und die wichtigsten commands</p> <ul style="list-style-type: none"> - Variablen, print(), input(), Type-Casts, f-Strings - Sequenzen, Verzweigungen und Schleifen - Funktionen definieren und aufrufen (positional / named parameter mit default values) - Externe Klassen und Module verwenden - Containers (Listen, Tuples und Dictionaries) - Filehandling - Sensoren und Aktoren vom Sense-Hat und GPIO in Python nutzen - REST Services nutzen und JSON Responses verarbeiten - Klassenkonzept in Python 		
Kompetenzen	A02.50	Informations- und Kommunikationstechnologien (ICT) professionell einsetzen und etablieren	
	A03.20	Neues Wissen mit geeigneten Methoden erschliessen und arbeitsplatznahe Weiterbildung realisieren	
	A03.30	Neue Technologien kritisch reflexiv beurteilen, adaptieren und integrieren	
	A03.40	Die eigenen digitalen Kompetenzen kontinuierlich weiterentwickeln	
	B10.10	Die Architektur der Software bestimmen und die Entwicklung unter Berücksichtigung von Betrieb und Wartung planen und dokumentieren	
	B11.30	Spezifikation in einer geeigneten Programmiersprache umsetzen	
	B11.40	Entwicklungsprojekte aufgrund der Analyseergebnisse und des gewählten Vorgehens planen und leiten	
	B11.50	Mobile und verteilte Applikationen unter Berücksichtigung zeitgemässer Architekturmuster bzw. Referenzarchitekturen implementieren	
	B11.60	Testkonzepte und Testspezifikation erstellen, Tests implementieren und auswerten sowie notwendige Massnahmen umsetzen	
Ziele	B11.80	Prinzipien, Methoden und Werkzeuge für die arbeitsteilige Entwicklung und Anwendung von umfangreichen Softwaresystemen zielorientiert bereitstellen und systematisch umsetzen	
	<p>K2 – Verstehen</p> <ol style="list-style-type: none"> 1. Sie können die grundlegenden UNIX-Befehle erklären und deren Bedeutung für die Python-Entwicklung beschreiben. 2. Sie können die Funktionsweise von GPIO-Pins und deren Einsatzmöglichkeiten auf dem Raspberry Pi erläutern. 3. Sie können die grundlegenden Konzepte von REST-Services und JSON-Format beschreiben. 4. Sie können den Aufbau einer Entwicklungsumgebung für Python auf einem BYOD-Gerät erklären. <p>K3 – Anwenden</p> <ol style="list-style-type: none"> 5. Sie können eine Python-Entwicklungsumgebung auf ihrem BYOD-Gerät einrichten und konfigurieren. 		

6. **Sie können** eine Verbindung zu einem Raspberry Pi über SSH und FTPS herstellen und nutzen.
7. **Sie können** einfache Python-Skripte schreiben, um Sensordaten von einem Sense-HAT auszulesen.
8. **Sie können** REST-Services in Python aufrufen und JSON-Daten verarbeiten, um sie in einer Anwendung zu verwenden.

K4 – Analysieren

9. **Sie können** API-Dokumentationen analysieren und daraus ableiten, wie externe Pakete in eigene Anwendungen integriert werden können.
10. **Sie können** bestehende Python-Skripte untersuchen und deren Funktionalität sowie Schwachstellen identifizieren.
11. **Sie können** die Anforderungen für eine Heizungssteuerung ableiten, basierend auf REST-API-Daten (z. B. Wettervorhersagen).
12. **Sie können** die Architektur einer Python-Anwendung analysieren, um Verbesserungspotenziale zu erkennen.

K5 – Bewerten

13. **Sie können** die Vor- und Nachteile von objektorientierter gegenüber funktionaler Programmierung in Python bewerten.
14. **Sie können** verschiedene Ansätze für die Steuerung von Aktoren auf einem Raspberry Pi kritisch vergleichen und Empfehlungen aussprechen.
15. **Sie können** die Effizienz und Sicherheit verschiedener Methoden für den Zugriff auf Raspberry Pi-Systeme beurteilen.
16. **Sie können** den Einsatz von Sense-HAT-Modulen in einem Projekt evaluieren und deren Nutzen für spezifische Anwendungen bewerten.

Voraussetzungen	Kennen einer anderen objektorientierten Sprache, wie Java, C#, C++
Lehrmittel	